Computers and Geosciences 118 (2018) 86–90

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Plumetrack: Flux calculation software for UV cameras

Nial Peters*,[1], Clive Oppenheimer[2]

Department of Geography, University of Cambridge, Downing Place, Cambridge, CB2 3EN, UK

A B S T R A C T

Ultraviolet (UV) cameras are increasingly employed to map and measure SO_2 abundances in volcanic emissions to the atmosphere. The main purpose of this is to estimate mass fluxes of SO_2, which requires estimation of the transport velocity of the plume. In this paper, we present Plumetrack, open-source Python based software for computing SO_2 fluxes from calibrated UV camera images. Designed to be the final component in UV camera processing toolchains, Plumetrack provides functionality for velocity estimation using optical-flow, flux calculation and error estimates. It can be used interactively via a graphical user interface or for batch processing via a commandline interface. We discuss the features and implementation details of Plumetrack, describe in detail a new flux calculation algorithm and demonstrate its performance on a set of synthetic UV camera images. The new algorithm is found to out perform the established flux calculation method, especially for highly spatio-temporally variable plumes. Furthermore, we show that the Plumetrack software may be successfully used with data from other imaging systems such as standard video cameras.

1. Introduction

SO_2 flux is an important parameter in the characterisation of volcanic activity. It is measured at numerous volcanoes worldwide as part of the operational monitoring campaigns of local observatories (e.g. Edmonds et al., 2003; Sweeney et al., 2008; Salerno et al., 2009) and is a major component of many research studies (e.g. Bani et al., 2012; Smekens et al., 2013; Pering et al., 2014; Carn et al., 2017; Moussallam et al., 2017). UV cameras (sometimes referred to as SO_2 cameras) are now a common tool for measuring volcanic SO_2 emissions, and with the recent development of low-cost units (Wilkes et al., 2016, 2017) are likely to become increasingly so. Using the difference in absorption of SO_2 at two different wavelengths (typically 310 nm and 330 nm), each pixel value in an UV camera image can be converted into an amount of SO_2 along the optical path of that pixel (typically expressed in units of ppm m or molec cm^{-2}). A detailed, though early, description of UV camera operation is given by Mori and Burton (2006). In 2015 Kern et al. (2015) performed a comparison of seven UV camera systems. A recent review is given by McGonigle et al. (2017). Although calibration of UV camera images is non-trivial (Kantzas et al., 2010; Kern et al., 2010, 2013; Lübcke et al., 2013) and is somewhat limited by the relatively wide bandpass of the filters used (typically 10 nm), their ability to image volcanic plumes in two dimensions is a great advantage compared with non-imaging ultraviolet spectroscopy (e.g. Oppenheimer, 2010). They are particularly useful for measurements at volcanoes where the plume does not rise coherently, making other spectroscopic measurement techniques difficult.

To compute fluxes from UV camera images, the velocity of the gas plume is required. This is typically estimated from the UV camera images by cross-correlating integrated column amount values from two complete transects of the gas plume (i.e., hypothetical surfaces that bisect the plume approximately perpendicularly to its direction of motion, and projected into the image plane) at different distances (parallel to the dominant direction of motion) from the source (McGonigle et al., 2005; Williams-Jones et al., 2006; Mori and Burton, 2006). However, this technique assumes a single velocity for the entire plume, often resulting in a significant over-estimate of the flux (Peters et al., 2014), and also requires a careful compromise between temporal- and spatial-resolution when performing the correlation (Boichu et al., 2010).

Recently, the potential of feature tracking algorithms that operate in two-dimensions has been realised for flux calculation purposes (Kern et al., 2012; Peters et al., 2014; Thomas and Prata, 2018). Such algorithms can compute a dense, 2D velocity field over the entire field of view of the camera, allowing differential motion across the plume to be accounted for. The benefits of 2D motion estimation algorithms for SO_2 flux calculations are discussed in more detail by Peters et al. (2014).

Here we present the Plumetrack program, open-source software for

* Corresponding author.
E-mail address: njp39@cam.ac.uk (N. Peters).
[1] Author contribution: Wrote and tested the software, authored initial text of the manuscript.
[2] Author contribution: Supervised the project, edited and revised the manuscript.

https://doi.org/10.1016/j.cageo.2018.05.014
Received 21 March 2018; Received in revised form 30 May 2018; Accepted 31 May 2018
Available online 04 June 2018

computing SO_2 fluxes from calibrated UV camera images, which uses a 2D optical flow algorithm to compute the plume motion between consecutive images. We also present a new integration algorithm implemented by Plumetrack, which is shown to be more robust for low-capture rate images (or highly spatiotemporally variable plumes). Example use cases and discussion of the limitations of the software are also provided. It is important to note that Plumetrack does not provide any of the functionality required to calibrate UV camera images. This must be done using other software, such as Vulcamera (Tamburello et al., 2011) or Pyplis (Gliβ et al., 2017), prior to processing the images with Plumetrack.

All results presented in this article were produced using version 15.06 of the Plumetrack software.

2. Software overview

The Plumetrack software is written primarily in Python, with an optional C++ extension module that allows use of CUDA-capable Graphics Processing Units (GPUs) to improve the computational speed of the motion estimation. Use of the Numpy, SciPy, Matplotlib and OpenCV libraries is made, and installation of these packages is required prior to using Plumetrack. Furthermore, pyinotify (for Linux users) and the Python win32 bindings (for Windows users) are also prerequisites. Full details of all these libraries along with links to their respective websites are given in the Plumetrack documentation, which is available at the link below. The graphical interface to the program is written using wxPython, however, Plumetrack can be used on systems without wxPython installed via its command line interface. Full source code, as well as release versions and accompanying documentation can be obtained under the terms of the Gnu Public License (version 3 or later) from https://github.com/nonbiostudent/plumetrack. Plumetrack has been successfully installed and tested on both Windows (7 and 10) and Linux (Ubuntu 16.04 LTS) using Python 2.7. It should also be possible to use it with other systems.

Motion estimation between successive images is achieved using OpenCV's implementation of the Farnebäck algorithm, a dense optical flow algorithm that has previously been shown to be effective at computing plume velocities (Peters et al., 2014). Plumetrack then offers two different algorithms for computing the flux across any user defined line. These algorithms are discussed in detail below.

The main functionality of the Plumetrack software is exposed via two different interfaces: a command-line interface and a graphical user interface (GUI). The command-line interface is designed to make it easy to incorporate Plumetrack into existing UV camera processing toolchains, and also to ease its use as an automated processing tool. The GUI, shown in Fig. 1, is designed to simplify use of the program for the majority of users. In addition to providing a simple “point and click” interface for running flux calculations, it also provides an interactive viewer tool, shown in Fig. 2, which allows the effects on the computed motion field of changing each configuration parameter to be viewed in realtime. This greatly simplifies selecting good configuration settings and allows for an empirical assessment of the sensitivity of the computed motion field to each parameter.

Correctness of the results produced by the software is ensured by a set of unit tests, created within the PyUnit testing framework. Although the test coverage is far from complete, it is being constantly improved upon and the flux algorithms are already well covered. The Farnebäck algorithm implementation itself is covered by the OpenCV test suite, and has been shown to produce good results for a set of synthetic UV camera images produced using a computer model (Peters et al., 2014).

Example UV camera images and Plumetrack configuration parameters are included as part of the source distribution for the software and provide a useful starting point for new users. Instructions for using these can be found in the “Quickstart” section of the Plumetrack documentation.

3. Flux algorithms

Once the plume velocity has been calculated, fluxes can be computed across any user defined path in the image, and multiple paths may be defined if desired. Henceforth, we refer to the user defined paths as integration lines. Plumetrack offers two different methods for computing the flux, which we refer to as the 1D (1-dimensional) and 2D (2-dimensional) algorithms.

3.1. 1D flux algorithm

The 1D flux algorithm is similar to the traditional method for computing fluxes from UV camera images, whereby each pixel value along the integration line is multiplied by the plume velocity perpendicular to the integration line at that location and these values are then summed and multiplied by the length of the integration line to give the flux. The Plumetrack 1D algorithm is identical to this, except that cubic-spline interpolation is used to compute pixel-values and velocities at positions other than on the image grid, allowing for complex integration line shapes. Only pixel values that fall on the integration line can contribute to the flux, and hence we refer to it as 1-dimensional. The potential weakness with this algorithm is that for a highly dynamic plume (for Example one which is separated into rapidly rising puffs), or for an image sequence with a long delay between images, it is conceivable that plume inhomogeneities that are on one side of the integration line in an image may have migrated to the other side in the subsequent image. Since they are never recorded on the integration line, they do not influence the flux computed by the 1D algorithm. Fig. 3 shows a simple cartoon of this. Although the figure is a simplification, it is intended merely to illustrate the failure mechanism of the 1D algorithm. A more rigorous demonstration of the algorithm's limitations is given in section 4.1.

3.2. 2D flux algorithm

The 2D flux algorithm overcomes the shortcomings of the 1D algorithm by considering the full velocity field computed by the Farnebäck algorithm. However, this comes at the expense of increased computation time (see Section 4.2). The algorithm works by considering the estimated motion vectors of every pixel in the image and computing which vectors intersect the integration line. Each pixel is then multiplied by its contribution factor: $+1$ or -1 if its motion vector crosses the integration line in a positive or negative direction respectively, 0 if it did not cross the integration line. All pixel values (weighted by their contribution factors) are then summed to obtain the flux. This method is less sensitive to large plume movements between successive images because all pixels are considered when computing the flux, rather than just those along the integration line, as in the 1D algorithm.

3.3. Error calculation

A simple estimate of the error in the computed fluxes may be made by considering conservation of mass. Given that for each image we know the SO_2 flux leaving each pixel (since we know its SO_2 mass and velocity), we can also compute the flux entering each pixel by considering all its neighbours. Hence, we can compute an estimate of what each pixel value should be in the subsequent image. By comparing this estimate with the recorded image we can evaluate the error associated with the flux estimate for each pixel. Using the standard error propagation formula, these individual errors may be combined to give an error estimate for our computed flux across the integration line.

Clearly, there are several quite poor assumptions associated with this calculation. Critically, it assumes that all the SO_2 recorded in a pixel moves as a homogeneous, discrete unit, which is rather unrealistic. Furthermore, no account is made of errors in the calibration of

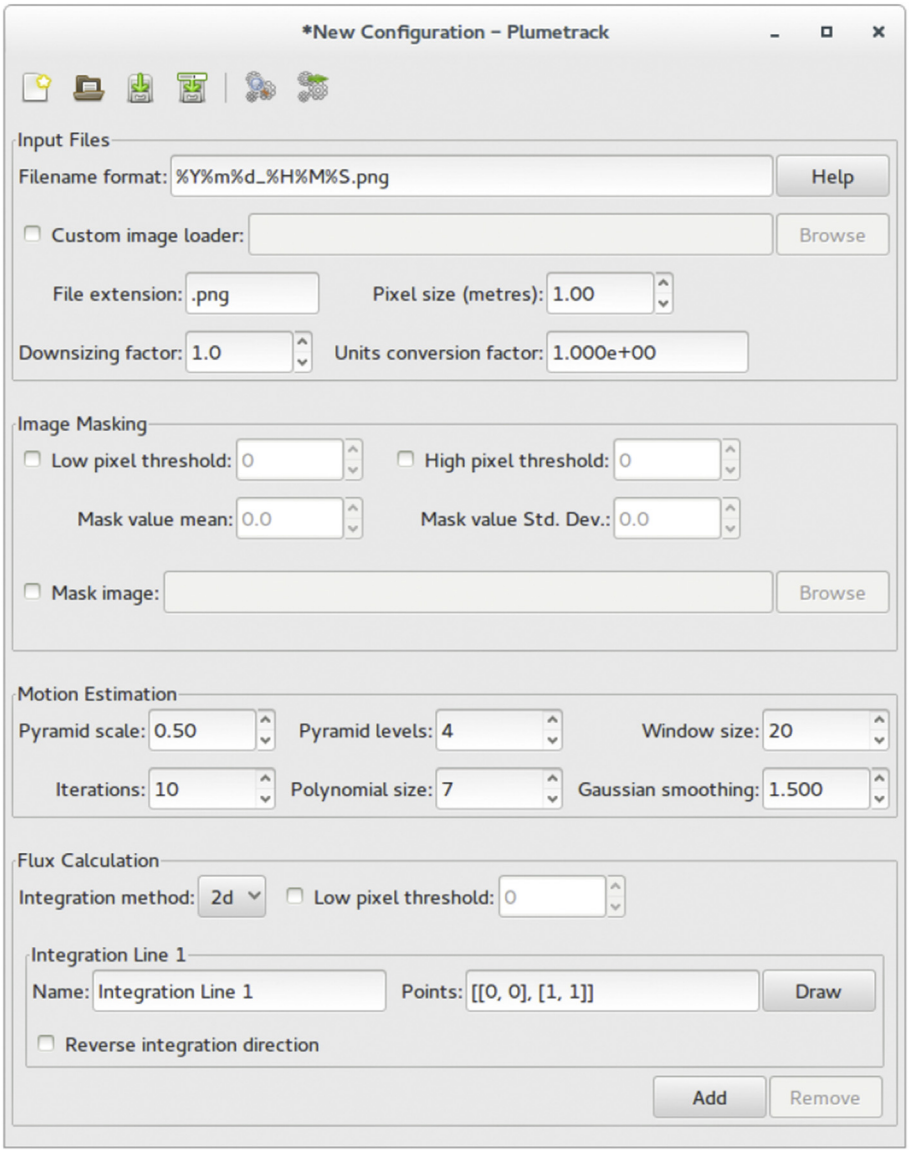

Fig. 1. Screenshot of the main Plumetrack GUI, changes to parameters here are reflected in the interactive viewer window (Fig. 2) in realtime.

the images and it should therefore not be treated as a robust estimate of error in the calculated flux. However, what the error computed by Plumetrack does provide is a self-consistent method of assessing how well the software is performing. Large estimated errors are indicative of poor quality results and as such the reported error can be used as a qualitative metric of the accuracy of the computed fluxes. Such information is of particular use in automated flux calculations, where large errors can be used to signal that user intervention may be required.

4. Results

4.1. Flux algorithm comparison

To better demonstrate the difference between the 1D and 2D flux calculation algorithms, we used Plumetrack to compute the SO_2 flux for a set of synthetic UV camera images. These images were taken from the "open-top" simulation dataset described in detail by Peters et al. (2014). The two algorithms were first applied to a series of images with a 1 s interval between them. Both algorithms produced very similar results, with errors of less than 1%. The image series was then subsampled, such that consecutive images had an 8 s interval between them. The results from applying Plumetrack to these images are shown in Fig. 4. Although the 2D algorithm continues to provide an accurate estimate of the flux, with errors of the order of 1%, the 1D algorithm shows deviations from the true flux value of up to 9%. This clearly demonstrates the superiority of the 2D algorithm for image series in which the plume moves a large amount between frames (either due to its rapid motion, or due to a low image capture rate).

4.2. Performance

On an 8-core 2.9 GHz Intel Core I7 system, using 512 × 512 pixel images and the 2D flux algorithm, Plumetrack can process 5 images per second. Using the 1D flux algorithm this increases to 8 images per second. Computation speed scales approximately linearly with the number of pixels in the images. To boost performance for large images, Plumetrack provides a downscaling option. Use of this option causes images to be resized before they are processed. The speed increase gained by use of this option comes at the expense of loss of resolution in the computed motion field. However, this loss of resolution can be beneficial for images with little small-scale structure, and using the downscaling option can be a good alternative to increasing the size of the averaging window for the Farnebäck algorithm.

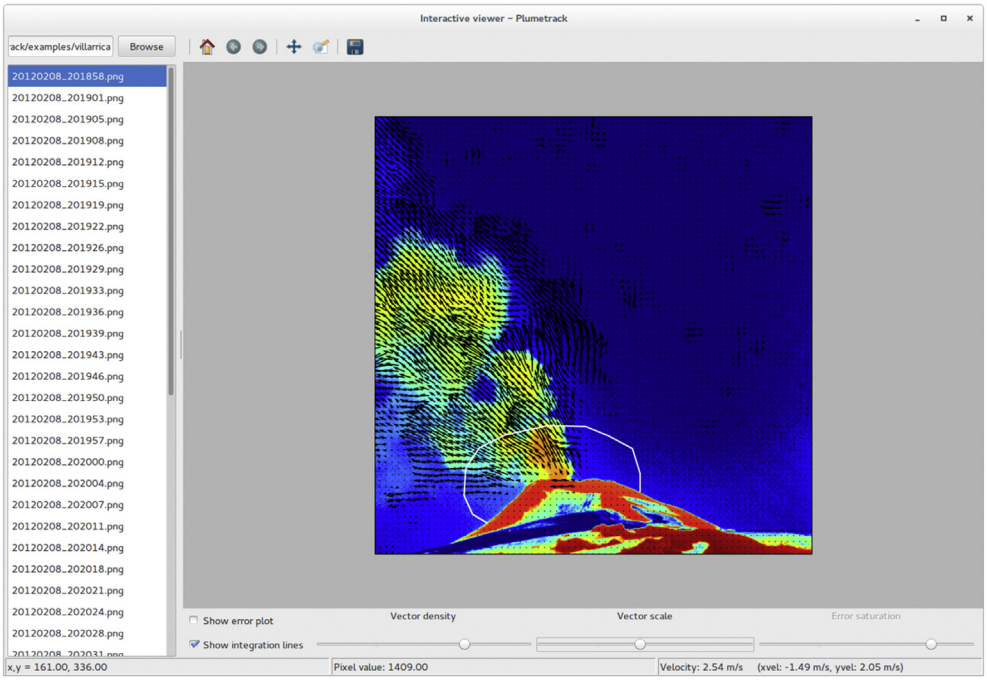

Fig. 2. Screenshot of the Plumetrack interactive viewer tool displaying one of the Example images (Villarrica volcano) provided as part of the Plumetrack distribution. Computed motion vectors are shown in black, and the user-defined integration line across which the flux is calculated is shown in white. Sliders at the bottom of the window control the appearance of the plot.

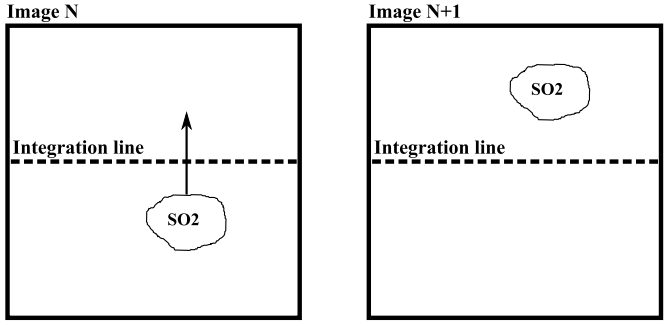

Fig. 3. Cartoon demonstrating the potential failure of the 1D algorithm to correctly compute the flux. The puff of high SO_2 concentration moves across the integration line between consecutive images and therefore does not contribute to the computed flux.

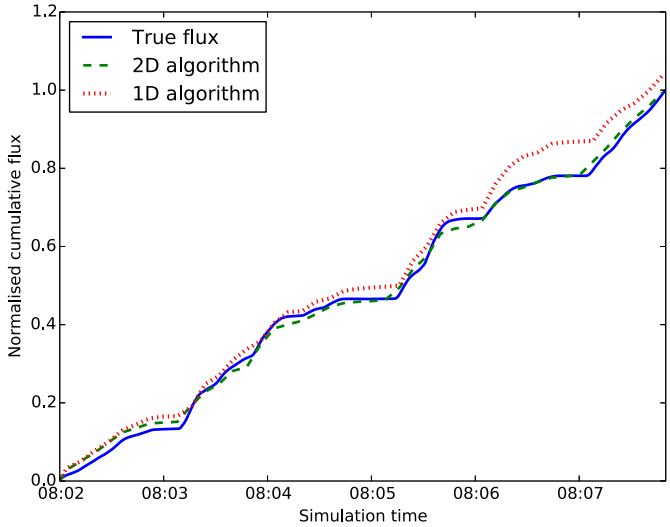

Fig. 4. Results of applying both the 1D and 2D flux algorithms to a set of synthetic UV camera images.

Use of GPU hardware can boost performance of the motion estimation, resulting in considerably faster processing times, especially for large images. However, in the current version of Plumetrack (15.06), use of the GPU is not compatible with parallel processing of multiple images. For a multi-core system therefore, use of the GPU is likely to degrade performance for batch processing of images (although it will improve performance for serial processing of images, for Example processing a real-time data stream from a camera). Parallel use of GPUs may be implemented in a future release of Plumetrack, but it is currently low priority.

4.3. Use of plumetrack with standard video capture

UV cameras are not the only method of measuring SO_2 flux from volcanoes. Indeed, scanning UV spectrometer systems are still prolific, and are used at numerous volcanoes around the world to monitor SO_2 emissions (e.g. Salerno et al., 2009). However, flux calculations for these systems still require the plume velocity to be known. This is usually taken to be either the windspeed, or is estimated manually from video footage of the plume. Although Plumetrack was designed to work with images from UV cameras, it provides sufficient flexibility to be used with images from any imaging system, providing a simple and fast way to compute the velocity of a plume. This can then be combined with UV spectrometer measurements to compute the flux. Fig. 5 shows a single frame from a video clip of the 2014 Holuhraun eruption in Iceland along with the motion field computed using Plumetrack. The optical flow algortihm used by Plumetrack requires grayscale images as its input. However, since the software allows users to define their own image loading functions, conversion to grayscale can be done when the image is loaded leaving the original image file unchanged.

Although designed with volcanic plume imaging in mind, Plumetrack may also be useful for other types of gas imaging systems (e.g. Gálfalk et al., 2016).

5. Summary and conclusions

Plumetrack is open-source software for computing SO_2 fluxes from UV camera images. It uses the Farnebäck optical flow algorithm to estimate the velocity of the SO_2 plume in the images, and offers two

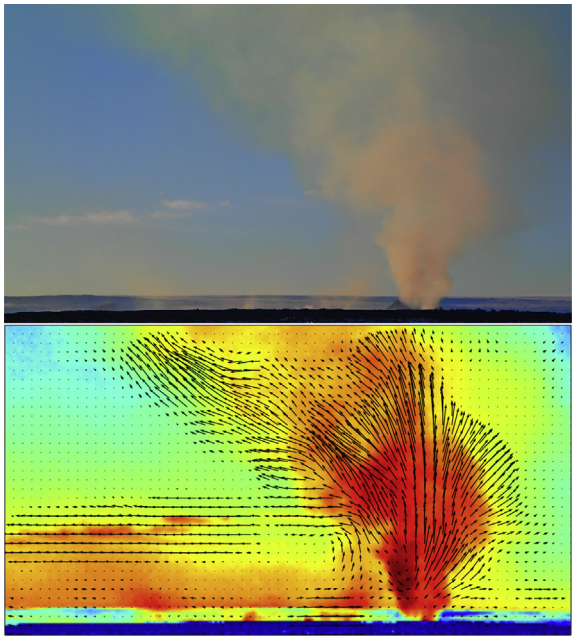

Fig. 5. Single frame of video of 2014 Holuhraun eruption (top) and corresponding motion field calculated using Plumetrack (bottom).

different algorithms for determining the flux. The 1D algorithm is computationally efficient, but is less accurate than the 2D algorithm for image sequences with large plume motions between frames. The software may be used either through a simple GUI or through a command-line interface. In addition to UV camera images, Plumetrack can also be used as a simple tool for determining the velocities of objects in any image sequence.

6. Computer code availability

The Plumetrack software, including documentation, Example data and full source code may be freely obtained from https://github.com/nonbiostudent/plumetrack.

Acknowledgements

This work was supported by the Isaac Newton Trust project "Physical constraints for the interpretation of open-vent volcanism" and NERC [grant NE/N009312/1]. CO is additionally supported by the NERC Centre for the Observation and Modelling of Volcanoes, Earthquakes and Tectonics (COMET).

Appendix A. Supplementary data

Supplementary data related to this article can be found at http://dx.doi.org/10.1016/j.cageo.2018.05.014.

References

Bani, P., Oppenheimer, C., Allard, P., Shinohara, H., Tsanev, V., Carn, S., Lardy, M., Garaebiti, E., 2012. First estimate of volcanic SO2 budget for Vanuatu island arc. J. Volcanol. Geoth. Res. 211, 36–46. http://dx.doi.org/10.1016/j.jvolgeores.2011.10.005.

Boichu, M., Oppenheimer, C., Tsanev, V., Kyle, P.R., 2010. High temporal resolution SO2 flux measurements at Erebus volcano, Antarctica. J. Volcanol. Geoth. Res. 190, 325–336. http://dx.doi.org/10.1016/j.jvolgeores.2009.11.020.

Carn, S.A., Fioletov, V.E., McLinden, C.A., Li, C., Krotkov, N.A., 2017. A decade of global volcanic SO_2 emissions measured from space. Sci. Rep. 7, 44095. http://dx.doi.org/10.1038/srep44095.

Edmonds, M., Herd, R.A., Galle, B., Oppenheimer, C.M., 2003. Automated, high time-resolution measurements of SO2 flux at Soufrière Hills Volcano, Montserrat. Bull. Volcanol. 65, 578–586. http://dx.doi.org/10.1007/s00445-003-0286-x.

Gålfalk, M., Olofsson, G., Crill, P., Bastviken, D., 2016. Making methane visible. Nat. Clim. Change 6, 426. http://dx.doi.org/10.1038/nclimate2877.

Gliß, J., Stebel, K., Kylling, A., Dinger, A.S., Sihler, H., Sudbø, A., 2017. Pyplis–a Python software toolbox for the analysis of SO2 camera images for emission rate retrievals from point sources. Geosciences 7, 134. http://dx.doi.org/10.3390/geosciences7040134.

Kantzas, E.P., McGonigle, A., Tamburello, G., Aiuppa, A., Bryant, R.G., 2010. Protocols for UV camera volcanic SO2 measurements. J. Volcanol. Geoth. Res. 194, 55–60. http://dx.doi.org/10.1016/j.jvolgeores.2010.05.003.

Kern, C., Kick, F., Lübcke, P., Vogel, L., Wöhrbach, M., Platt, U., 2010. Theoretical description of functionality, applications, and limitations of SO_2 cameras for the remote sensing of volcanic plumes. Atmos. Meas. Tech. Discuss. 3, 531–578. http://dx.doi.org/10.5194/amtd-3-531-2010.

Kern, C., Lübcke, P., Bobrowski, N., Campion, R., Mori, T., Smekens, J.F., Stebel, K., Tamburello, G., Burton, M., Platt, U., Prata, F., 2015. Intercomparison of SO2 camera systems for imaging volcanic gas plumes. J. Volcanol. Geoth. Res. 300, 22–36. http://dx.doi.org/10.1016/j.jvolgeores.2014.08.026.

Kern, C., Werner, C., Elias, T., Sutton, A.J., Lübcke, P., 2013. Applying UV cameras for SO2 detection to distant or optically thick volcanic plumes. J. Volcanol. Geoth. Res. 262, 80–89. http://dx.doi.org/10.1016/j.jvolgeores.2013.06.009.

Kern, C., Werner, C.A., Doukas, M.P., Elias, T., Kelly, P.J., Sutton, A.J., 2012. Imaging volcanic SO2 plumes with UV cameras. In: EGU General Assembly Conference Abstracts. EGU, Vienna, pp. 12596. http://adsabs.harvard.edu/abs/2012EGUGA.1412596K.

Lübcke, P., Bobrowski, N., Illing, S., Kern, C., Alvarez Nieves, J.M., Vogel, L., Zielcke, J., Delgado Granados, H., Platt, U., 2013. On the absolute calibration of SO2 cameras. Atmos. Meas. Technol. 6, 677–696. http://dx.doi.org/10.5194/amt-6-677-2013.

McGonigle, A.J.S., Hilton, D.R., Fischer, T.P., Oppenheimer, C., 2005. Plume velocity determination for volcanic SO2 flux measurements. Geophys. Res. Lett. 32http://dx.doi.org/10.1029/2005GL022470. L11302.

McGonigle, A.J.S., Pering, T.D., Wilkes, T.C., Tamburello, G., Aleo, R., Bitetto, M., Aiuppa, A., Willmott, J.R., 2017. Ultraviolet imaging of volcanic plumes: a new paradigm in volcanology. Geosciences 7.

Mori, T., Burton, M., 2006. The SO2 camera: a simple, fast and cheap method for ground-based imaging of SO2 in volcanic plumes. Geophys. Res. Lett. 33http://dx.doi.org/10.1029/2006GL027916. L24804.

Moussallam, Y., Tamburello, G., Peters, N., Apaza, F., Schipper, C.I., Curtis, A., Aiuppa, A., Masias, P., Boichu, M., Bauduin, S., Barnie, T., Bani, P., Giudice, G., Moussallam, M., 2017. Volcanic gas emissions and degassing dynamics at Ubinas and Sabancaya volcanoes; implications for the volatile budget of the central volcanic zone. J. Volcanol. Geoth. Res. 343, 181–191. http://dx.doi.org/10.1016/j.jvolgeores.2017.06.027.

Oppenheimer, C., 2010. Ultraviolet sensing of volcanic sulfur emissions. Elements 6, 87–92. http://dx.doi.org/10.2113/gselements.6.2.87.

Pering, T.D., Tamburello, G., McGonigle, A.J.S., Aiuppa, A., Cannata, A., Giudice, G., Patanè, D., 2014. High time resolution fluctuations in volcanic carbon dioxide degassing from Mount Etna. J. Volcanol. Geoth. Res. 270, 115–121. http://dx.doi.org/10.1016/j.jvolgeores.2013.11.014.

Peters, N., Hoffmann, A., Barnie, T., Herzog, M., Oppenheimer, C., 2014. Use of motion estimation algorithms for improved flux measurements using SO2 cameras. J. Volcanol. Geoth. Res. http://dx.doi.org/10.1016/j.jvolgeores.2014.08.031.

Salerno, G., Burton, M., Oppenheimer, C., Caltabiano, T., Randazzo, D., Bruno, N., Longo, V., 2009. Three-years of SO2 flux measurements of Mt. Etna using an automated UV scanner array: comparison with conventional traverses and uncertainties in flux retrieval. J. Volcanol. Geoth. Res. 183, 76–83. http://dx.doi.org/10.1016/j.jvolgeores.2009.02.013.

Smekens, J., Burton, M.R., Clarke, A.B., Harijoko, A., Wibowo, H., Sawyer, G., 2013. High frequency SO2 flux measurements at Semeru volcano, Indonesia, using the SO2 camera. In: AGU Fall Meeting Abstracts -1, 2882.

Sweeney, D., Kyle, P.R., Oppenheimer, C., 2008. Sulfur dioxide emissions and degassing behavior of Erebus volcano, Antarctica. J. Volcanol. Geoth. Res. 177, 725–733. http://dx.doi.org/10.1016/j.jvolgeores.2008.01.024.

Tamburello, G., Kantzas, E.P., McGonigle, A.J.S., Aiuppa, A., 2011. Vulcamera: a program for measuring volcanic SO2 using UV cameras. Ann. Geophys. 54. http://dx.doi.org/10.4401/ag-5181.

Thomas, H.E., Prata, A.J., 2018. Computer vision for improved estimates of SO2 emission rates and plume dynamics. Int. J. Rem. Sens. 39, 1285–1305. http://dx.doi.org/10.1080/01431161.2017.1401250.

Wilkes, T.C., McGonigle, A.J.S., Pering, T.D., Taggart, A.J., White, B.S., Bryant, R.G., Willmott, J.R., 2016. Ultraviolet imaging with low cost smartphone sensors: development and application of a raspberry pi-based UV camera. Sensors 16, 1649. http://dx.doi.org/10.3390/s16101649.

Wilkes, T.C., Pering, T.D., McGonigle, A.J.S., Tamburello, G., Willmott, J.R., 2017. A low-cost smartphone sensor-based UV camera for volcanic SO2 emission measurements. Rem. Sens. 9, 27. http://dx.doi.org/10.3390/rs9010027.

Williams-Jones, G., Horton, K.A., Elias, T., Garbeil, H., Mouginis-Mark, P.J., Sutton, A.J., Harris, A.J.L., 2006. Accurately measuring volcanic plume velocity with multiple UV spectrometers. Bull. Volcanol. 68, 328–332. http://dx.doi.org/10.1007/s00445-005-0013-x.